

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number  
**WO 01/71668 A2**

(51) International Patent Classification<sup>7</sup>: **G06T 15/20**

20 Comstock Queen Court, Mountain View, CA 94043  
(US). **MALZBENDER, Thomas** [US/US]; 4331 Miranda  
Avenue, Palo Alto, CA 94306 (US).

(21) International Application Number: **PCT/US01/07357**

(74) Agent: **SCHUYLER, Marc, P.**; Hewlett-Packard Company, Legal Dept., 3403 E. Harmony Road, Fort Collins, CO 80528-9599 (US).

(22) International Filing Date: 6 March 2001 (06.03.2001)

(81) Designated State (*national*): JP.

(25) Filing Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

(26) Publication Language: English

Published:

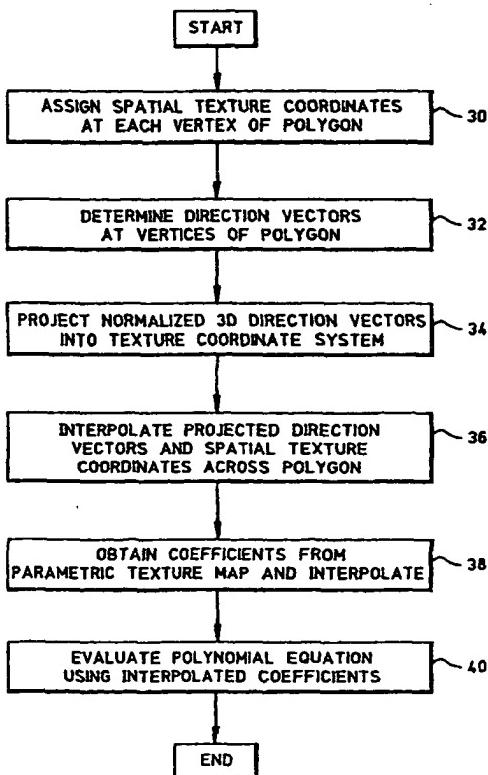
(30) Priority Data:  
09/528,700 17 March 2000 (17.03.2000) US

— without international search report and to be republished upon receipt of that report

(71) Applicant: **HEWLETT-PACKARD COMPANY**  
[US/US]; 1501 Page Mill Road, M/S 4U-10, Palo Alto,  
CA 94304-1126 (US).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: APPARATUS FOR AND METHOD OF RENDERING 3D OBJECTS WITH PARAMETRIC TEXTURE MAPS



(57) Abstract: A graphics system that employs parametric texture maps. The graphics system renders surface features of a 3D object in a manner that is direction dependent but without the time consuming and expensive calculations involved in the evaluation of lighting equations on a per pixel basis. A parametric texture map holds a set of parameters that define a surface structure in a manner in which the appearance of the surface structure varies in response to a direction vector. The direction vector may be any user-defined vector including a light source vector or a half-angle vector. The parameters are those of a predetermined equation, the evaluation of which does not involve vector calculations. The equation may take any form including a polynomial equation or a non-polynomial equation. The graphic system renders a polygon with the surface structure using the equation.

WO 01/71668 A2

**APPARATUS FOR AND METHOD OF RENDERING 3D OBJECTS  
WITH PARAMETRIC TEXTURE MAPS**

**BACKGROUND OF THE INVENTION**

5      1.     **Field of the Invention**

The present invention relates generally to the field of computer graphics and pertains more particularly to an apparatus for and method of rendering three-dimensional (3D) objects with parametric texture maps in a computer graphics system.

10     2.     **Discussion of the Prior Art**

Modern computer systems have become increasingly graphics intensive. Dedicated special purpose memories and hardware have been developed to meet this need. A conventional computer graphics system includes a display device having a two-dimensional (2D) array of light emitting areas. The light emitting areas are usually referred to as pixels which is an abbreviation for picture elements. Such a graphics system typically employs 15 hardware, software, or both to generate a 2D array of values that determine the colors or shades of grey that are to be emitted from the corresponding pixels of the display.

Computer graphics systems are commonly employed for the display of 3D objects. Typically, such a system generates what appears to be a 3D object on a 2D display by generating 2D views of the 3D object that is modeled in the computer memory. The 2D view 20 of a 3D object which is generated at a particular time usually depends at least on a spatial relationship between the 3D object and a viewer of the 3D object at the particular time. This spatial relationship may be referred to as the view or eye point direction. For example, a car is a 3D object having a front and a back. However, whether one can see the taillights will depend on where one is viewing the car from. A view direction that is directly in front of the 25 car will not show the taillights while a view direction that is directly behind the car will.

The process by which a computer graphics system generates the values for a 2D view of a 3D object is commonly referred to as image rendering or scan conversion. The graphics system usually renders a 3D object by subdividing the 3D object into a set of polygons and rendering each of the polygons individually.

5       The values for a polygon that are rendered for a particular view direction usually depend on the surface features of the polygon and the effects of lighting on the polygon. The surface features include details such as surface colors and surface structures. The effects of lighting usually depend on a spatial relationship between the polygon and one or more light sources. This spatial relationship may be referred to as the light source direction. For  
10      example, if there is only one light source, the side of the object closest to the light source will be illuminated while the side of the object furthest from the light source might be in shadow.

Typically, the evaluation of the effects of lighting on an individual pixel in a polygon for a particular view direction involves a number of 3D vector calculations. One of ordinary skill in the art will recognize that the standard Blinn/Phong lighting equation is as follows:

15       
$$I = k_a I_a + k_d I_d (N \cdot L) + k_s I_s (N \cdot H)^n \quad (1)$$

where  $k_a$ ,  $k_d$ , and  $k_s$ , are constants. Equation (1) states that the light intensity  $I$  for a particular pixel is a function of the sum of the ambient contribution  $I_a$ , the diffuse contribution  $I_d$ , and the specular contribution  $I_s$ , at that location. Lighting calculations based on Equation (1) include floating-point, square-root and divide operations when used with normalized vectors.  
20      Such calculations are usually time consuming and expensive whether performed in hardware or software.

One conventional method for reducing such computation overhead is to evaluate the effects of lighting at just a few areas of a polygon, such as the vertices, and then to interpolate the results across the entire polygon. Examples include methods which are commonly referred to as flat shading and Gouraud shading. Such methods usually reduce the number of  
25

calculations that are performed during scan conversion and thereby increase rendering speed. Unfortunately, such methods also usually fail to render shading features that are smaller than the areas of individual polygons. If the polygons are relatively large, the view will be noticeably distorted.

5 One conventional method for rendering features that are smaller than the area of a polygon is to employ what is commonly referred to as a texture map. A typical texture map is a table that contains a pattern of color values for a particular surface feature. For example, a wood grain surface feature may be rendered using the surface and a texture map that holds a color pattern for wood grain. Unfortunately, texture mapping usually yields relatively flat  
10 surface features that do not change with the view direction or light source direction. The appearance of real 3D objects, on the other hand, commonly do change with the view direction, light source direction, or both. These directional changes are commonly caused by 3D structures on the surface of the object, that is, the object is not perfectly flat. Such structures can cause localized shading or occlusions or changes in specular reflections from a  
15 light source. The effects can vary with view direction for a given light source direction and can vary with light source direction for a given view direction. These directional changes should be accounted for to provide greater realism in the rendered 2D views.

One conventional method for handling the directional dependence of such structural effects in a polygon surface is to employ what is commonly referred to as a bump map. A  
20 typical bump map contains a height field from which a pattern of 3D normal vectors for a surface are extracted. The normal vectors are used to evaluate lighting equations at each pixel in the surface. Unfortunately, such evaluations typically involve a number of expensive and time consuming 3D vector calculations including division and square roots. This can result in decreased rendering speed or increased graphics system cost.

A definite need exists for a system having an ability to meet the efficiency requirements of graphics intensive computer systems. In particular, a need exists for a system which is capable of rendering 2D views of a 3D object in a skillful manner. Ideally, such a system would have a lower cost and a higher productivity than conventional systems.

- 5 With a system of this type, system performance can be enhanced. A primary purpose of the present invention is to solve this need and provide further, related advantages.

### SUMMARY OF THE INVENTION

A graphics system is disclosed that employs parametric texture maps. The graphics 10 system renders surface features of a 3D object in a manner that is direction dependent but without the time consuming and expensive calculations involved in the evaluation of lighting equations on a per pixel basis. A parametric texture map holds a set of parameters that define a surface structure in a manner in which the appearance of the surface structure varies in response to a direction vector. The direction vector may be any user-defined vector including 15 a light source vector or a half-angle vector. The parameters are those of a predetermined equation, the evaluation of which does not involve vector calculations. The equation may take any form including a polynomial equation or a non-polynomial equation. The graphic system renders a polygon with the surface structure using the equation.

**BRIEF DESCRIPTION OF THE DRAWING**

The above and other objects and advantages of the present invention will be more readily appreciated from the following detailed description when read in conjunction with the accompanying drawing, wherein:

5 FIG. 1 is a block diagram of a computer graphics system according to a preferred embodiment of the present invention;

FIG. 2 is a graphical representation of a polygon which is defined in the buffer of FIG. 1 and which is to be rendered by the graphics processor of FIG. 1 using surface features defined by the parametric texture map of FIG. 1;

10 FIG. 3 is a flow diagram of a method according to the present invention for rendering a polygon using surface features defined by a parametric texture map; and

FIGS. 4A-4C illustrate an arrangement for determining the coefficients in a parametric texture map for an example surface structure employed in the present invention.

15 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

Turning first to FIG. 1, a block diagram of a computer graphics system 10 according to a preferred embodiment of the present invention is shown. The computer graphics system 10 includes a buffer 12, a graphics processor 14, a parametric texture map 16, a frame buffer 18, and a display 20. The buffer 12 holds geometry data that describes a 3D object which is 20 to be generated on the display 20. The 3D object is represented in the buffer 12 as a set of polygons in a 3D space. In one embodiment, the polygons are triangles and the geometry data in the buffer 12 includes the 3D coordinates of the vertices of the triangles.

The graphics processor 14 reads the parameters that define the polygons from the buffer 12 and scan converts each polygon. The scan conversion of a polygon yields a 2D 25 view of the polygon which depends on a view direction and a light source direction. A 2D

view of a polygon includes a color value for each pixel of the polygon which is visible in the plane of the display 20. The graphics processor 14 writes the color values for the rendered polygons into the frame buffer 18. The color values from the frame buffer 18 are provided to the display 20 on a frame by frame basis. The display 20 can be any conventional 2D display device such as a raster scan device or flat-panel display device.

The parametric texture map 16 holds parameters that define a surface structure in a manner in which the appearance of the surface structure varies with any user-defined vector such as the view vector, the light source vector, or the half-angle vector. The half-angle vector is a vector that is halfway between the view and light vectors. The graphics processor 14 maps the surface structure defined in the parametric texture map 16 onto the polygons obtained from the buffer 12 during scan conversion. The result is a more realistic rendering of 3D features in a surface on a 3D object in comparison to texture mapping but without the computational penalties associated with bump mapping.

In one embodiment, the parameters contained in the parametric texture map 16 are the  $A1, A2, A3, A4, A5$ , and  $A6$  coefficients for evaluating the following second order polynomial equation:

$$C_i = A1D_u^2 + A2D_v^2 + A3D_uD_v + A4D_u + A5D_v + A6 \quad (2)$$

where  $D_u$  and  $D_v$ , are the 2D components of the user-defined vector. For example, if the parametric texture map 16 is adapted to the view direction then  $D_u$  and  $D_v$  are the 2D components of an eye point vector. Likewise, the terms  $D_u$  and  $D_v$  are the 2D components of a half-angle vector if the parametric texture map 16 is adapted to the half-angle vector.

TABLE 1 illustrates the general contents of the parametric texture map 16. The parametric texture map 16 contains  $n$  by  $m$  entries. Each of the  $n$  by  $m$  entries corresponds to a sample of a particular surface modeled by the parametric texture map 16. These samples may be referred to as texels which is an abbreviation for texture elements. The coefficients

for an individual texel are denoted as  $A1_{i,j}$  to  $A6_{i,j}$  wherein  $i$  ranges from one to  $n$  and  $j$  ranges from one to  $m$ .

TABLE 1

$A1_{1,1}$ $A2_{1,1}$	$A1_{1,2}$ $A2_{1,2}$	...	$A1_{1,m}$ $A2_{1,m}$
$A3_{1,1}$ $A4_{1,1}$	$A3_{1,2}$ $A4_{1,2}$		$A3_{1,m}$ $A4_{1,m}$
$A5_{1,1}$ $A6_{1,1}$	$A5_{1,2}$ $A6_{1,2}$		$A5_{1,m}$ $A6_{1,m}$
$A1_{2,1}$ $A2_{2,1}$	$A1_{2,2}$ $A2_{2,2}$		$A1_{2,m}$ $A2_{2,m}$
$A3_{2,1}$ $A4_{2,1}$	$A3_{2,2}$ $A4_{2,2}$	...	$A3_{2,m}$ $A4_{2,m}$
$A5_{2,1}$ $A6_{2,1}$	$A5_{2,2}$ $A6_{2,2}$		$A5_{2,m}$ $A6_{2,m}$
.	.		.
.	.		.
.	.		.
$A1_{n,1}$ $A2_{n,1}$	$A1_{n,2}$ $A2_{n,2}$		$A1_{n,m}$ $A2_{n,m}$
$A3_{n,1}$ $A4_{n,1}$	$A3_{n,2}$ $A4_{n,2}$	...	$A3_{n,m}$ $A4_{n,m}$
$A5_{n,1}$ $A6_{n,1}$	$A5_{n,2}$ $A6_{n,2}$		$A5_{n,m}$ $A6_{n,m}$

- 5        The parametric texture map 16 is representative of a set of parametric texture maps  
 that may be used for rendering 3D objects in the graphics system 10. Each parametric texture  
 map according to the present technique is adapted to a particular surface structure that is to be  
 mapped onto a 3D object. In addition, each parametric texture map is adapted to provide  
 realistic 3D rendering in response to the user-defined vector. For example, the parametric  
 10      texture map 16 may be adapted to provide realistic 3D rendering in response to a varying  
 light source direction for a given fixed view direction. Alternatively, the parametric texture

map 16 may be adapted to provide realistic 3D rendering in response to a varying view direction for a given fixed light source direction.

In one embodiment, a parametric texture map may be adapted to a particular color channel of the display 20. For example, the graphic system 10 may include a separate 5 parametric texture map for each of the red, green, and blue channels for an RGB display for a particular surface structure. One drawback to this approach is that three parametric texture maps are required. Alternatively, a single parametric texture map may be multiplied with each channel in a complete RGB texture map. In this embodiment, the parametric texture map encodes the brightness changes as a result of changes in the light vector, for example, 10 and modulates the RGB color values in a corresponding texture map to yield the resultant color values. It is also possible to render objects illuminated by multiple light sources using this type of parametric texture map. Since the RGB color values are separate from the intensity, the intensity values can be calculated for different light sources and combined to get a final intensity value. The per-pixel intensity values are then multiplied with the RGB 15 values to get the output color. A wide range of useful operations can be achieved by using different blending modes to combine parametric texture map evaluation results with calculated lighting, texture maps, or other sources during texture blending in the graphics pipeline.

Turning now to FIG. 2, a graphical representation of a polygon 22 which is defined in 20 the buffer 12 of FIG. 1 and which is to be rendered by the graphics processor 14 of FIG. 1 using surface features defined by the parametric texture map 16 of FIG. 1 is shown. The polygon 22 is defined by a set of three vertices ( $T_1$ ,  $T_2$ , and  $T_3$ ) in a 3D space. The local coordinate space is represented by a set of  $u$  and  $v$  axes 24 which are usually defined by the spatial texture coordinates of the polygon 22.

A surface normal vector  $N$  for the polygon 22 is shown, along with an eye point vector  $V$ , a light source vector  $L$ , and a half-angle vector  $H$ . Note that normal vectors are usually defined for the vertices rather than the entire polygon to improve realism. Here a single normal for the polygon is illustrated for simplicity. The eye point vector  $V$  represents a view direction from a pixel  $P_k$  of the polygon 22 to an eye point 26. The light source vector  $L$  represents a light source direction from the pixel  $P_k$  to a light source 28. The half-angle vector  $H$  represents the vector that is halfway between the eye point vector  $V$  and the light source vector  $L$ . Also shown is a  $(V_u, V_v)$  vector which is the eye point vector  $V$  projected down into the plane of the polygon 22. Likewise, a  $(L_u, L_v)$  vector is the projection of the light source vector  $L$  and a  $(H_u, H_v)$  vector is the projection of the half-angle vector  $H$ .

FIG. 3 shows a flow diagram of a method according to the present invention for rendering a polygon using surface features defined by a parametric texture map. The steps shown are used to generate a color value for each of a set of pixels in the polygon 22 of FIG. 2. The following description for purposes of illustration focuses on the pixel  $P_k$  of FIG. 2 as an example. The coefficients  $A1_{ij}$  to  $A6_{ij}$  in the parametric texture map 16 of FIG. 1 are adapted to yield color values in response to a user-defined vector such as a light source vector or an half-angle vector. This adaptation is done by using linear regression from a set of images that provide color values at each pixel for varying vector directions. The following description for purposes of illustration focuses on an example in which the coefficients  $A1_{ij}$  to  $A6_{ij}$  are adapted to yield color values in response to a light source vector for a fixed eye point vector. Nevertheless, these techniques are readily applicable to a parametric texture map which contains coefficients that are adapted to yield color values in response to any user-defined vector. In addition, the coefficients  $A1_{ij}$  to  $A6_{ij}$  in the parametric texture map 16 yield color values for a particular color channel of the display 20 of FIG. 1. Additional parametric texture maps may be used to yield color values for the remaining channels. The

steps are repeated with each polygon in the set of polygons representing the 3D object that is stored in the buffer 12 of FIG. 1. The process begins at START.

At step 30, the graphics processor 14 of FIG. 1 assigns spatial texture coordinates  $u$  and  $v$  of the parametric texture map 16 at each vertex  $T_1$ ,  $T_2$ , and  $T_3$  of the polygon 22. The 5 spatial texture coordinates for the vertices  $T_1$ ,  $T_2$ , and  $T_3$  are denoted as  $u_{T1}$ ,  $v_{T1}$ ,  $u_{T2}$ ,  $v_{T2}$ , and  $u_{T3}$ ,  $v_{T3}$ , respectively.

At step 32, the graphics processor 14 determines direction vectors at the vertices of the polygon 22. The direction vectors in this example are light source vectors at the vertices of the polygon 22. The light source vector at the vertex  $T_1$  is a normalized 3D vector that 10 points from  $T_1$  to the 3D coordinates of the light source 28 of FIG. 2. Similarly, the light source vector at the vertices  $T_2$  and  $T_3$  are normalized 3D vectors that point from  $T_2$  and  $T_3$ , respectively, to the 3D coordinates of the light source 28.

At step 34, the graphics processor 14 projects the normalized 3D direction vectors determined at step 32 into the texture coordinate system  $u$  and  $v$  of the parametric texture map 15 16. This yields 2D parameterization or 2D components of each normalized 3D direction vector in the texture coordinate system of the parametric texture map 16. A 2D parameterization of a normalized 3D direction vector is denoted as  $D_u$ ,  $D_v$ .

At step 36, the graphics processor 14 interpolates the projected direction vectors  $D_u$ ,  $D_v$ , determined at step 34 and spatial texture coordinates  $u_{T1}$ ,  $v_{T1}$ ,  $u_{T2}$ ,  $v_{T2}$ , and  $u_{T3}$ ,  $v_{T3}$  20 determined at step 30 across the polygon 22. This associates each pixel of the polygon 22 with  $D_u$ ,  $D_v$  parameters and with  $u$  and  $v$  texel coordinates in the coordinate space of the parametric texture map 16. The interpolation of step 36 may be performed using a variety of known techniques.

At step 38, the graphics processor 14 obtains the polynomial coefficients  $A1_{i,j}$  to  $A6_{i,j}$  25 from the parametric texture map 16 and interpolates them. It is important to note that there

are several options for evaluating and interpolating the texture map. Interpolated texture coordinates will not yield exact texels in the input texture map so texture interpolation is needed. These interpolation methods are well known and include nearest-neighbor, bilinear, trilinear, and anisotropic interpolation. One option is to evaluate the polynomial at each of 5 the sample points and then to interpolate the output value. Another option is to interpolate the polynomial coefficients to generate an interpolated polynomial and then to evaluate the polynomial. FIG. 3 illustrates the second option.

At step 40, the graphics processor 14 evaluates Equation (2) using the interpolated  $D_u$  and  $D_v$  terms from step 36 and the interpolated coefficients  $A1_{i,j}$  to  $A6_{i,j}$  from step 38 on a per 10 pixel basis. For pixel  $P_k$ , Equation (2) yields a texel value  $C_i$  which may then be used as a color value, luminance value, or other pixel or texture data.

It is generally the case that parametric texture mapping effects should only be applied to polygons that are front facing relative to the user-defined vector being used. In order to identify front facing and back facing polygons, a third variable is used. This third parameter 15  $D$  is typically the result of projecting the user-defined vector onto the vertex normal vectors. For front facing polygons,  $D$  will be positive and for back facing polygons,  $D$  will be negative. Fundamentally,  $D$  is the dot product between the vertex normal vector and the user-defined vector. It may or may not be desirable to have the parametric texture mapping effects stop abruptly at the transition from front facing to back facing. If desirable, the user 20 can attenuate the effects to give a sharp or gradual cutoff as needed, possibly including evaluation for triangles that are back facing with respect to the user-defined vector.

The present techniques model the 3D effects of surface features by modeling the contribution of those features to surface colors directly. The contributions are then represented by the coefficients of Equation (2). The present techniques yield a mapping of 25 surface features to a polygon that is direction dependent and that provides a quality of realism

which is comparable to that yielded by the evaluation of lighting equations on a per pixel basis. Yet the evaluation of Equation (2), a second order polynomial, is relatively easy to perform in hardware, software, or both in comparison to the evaluation of lighting equations on a per pixel basis. The evaluation of Equation (2) involves integer multiply and add operations whereas the evaluation of lighting equations involves floating-point, square-root and divide operations. Although Equation (2) has been chosen to be a bi-quadratic polynomial, bi-cubic and higher order polynomials could also be used. Even non-polynomial functions could be supported by the present techniques. However, more complex functions are not currently preferred due to their higher implementation costs which are currently considered excessive.

Turning now to FIGS. 4A-4C, an arrangement for determining the coefficients  $AJ_{i,j}$  to  $A6_{i,j}$  of the parametric texture map 16 of FIG. 1 for an example surface structure 42 is illustrated. FIG. 4A shows a top view of the surface structure 42. The surface structure 42 is shown aligned to a pair of axes  $D_u$  and  $D_v$ . The surface structure comprises an array of square right pyramid structures 44-52. A sub area 54 represents one of the texels of the surface structure. The surface structure is just one example of a surface structure and any imaginable surface structure may be modeled using the present techniques.

FIG. 4B shows a side view of the surface structure 42 of FIG. 4A along with a hemispherical dome 56 which is used to position a camera 58 and a light source. In this case, a single light source is shown positioned at two example positions 60 and 62 on the dome. A vector 64 represents a light source vector for the position 60 and a vector 66 represents a light source vector for the position 62.

The camera 58 is fixed in its position on the dome 56 when obtaining coefficients for a parametric texture map 16 that is adapted to yield color values in response to the light source vector. A vector 68 represents the eye point vector for the fixed position of the

camera. The camera is used to obtain an image of the surface structure 42 for each of a set of predetermined positions of the light source on the dome 56. Each predetermined position of the light source represents a different light source vector and the corresponding image obtained with the camera yields a color value for each texel of the surface structure 42. For example, images obtained from  $n$  different positions of the light source yields  $n$  color values for each texel with each color value corresponding to a different light source vector for the eye point vector 68.

FIG. 4C shows a graph of the color values for the texel 54 obtained with the camera 58 for six different positions of the light source on the dome 56. These color values include a color value 60' which was obtained with the light source at position 60 and a color value 62' which was obtained with the light source at position 62. The color values are plotted against the axes  $D_u$  and  $D_v$ . The coefficients  $A1_{i,j}$  to  $A6_{i,j}$  for the texel 54 for the eye point vector 68 are obtained by fitting the polynomial of Equation (2) to the color values obtained for the texel 54 using known techniques. A surface 70 is shown that represents a fit between the polynomial of Equation (2) and the six color values obtained by the camera 58 for the texel 54. The fit of the surface 70 to the  $n$  color values obtained by the camera 58 may be accomplished using standard least mean square methods to yield the polynomial coefficients.

This technique may readily be modified to obtain coefficients for a parametric texture map 16 that is adapted to yield color values in response to any user-defined vector. For example, the light source may be fixed at position 60 which corresponds to the light source vector 64. Then the camera 58, representing the eye point vector, is moved to  $n$  different positions on the dome 56. A surface fit to the obtained color values is performed in a manner similar to that described above.

A further modification of this technique is to combine parametric texture maps with existing per-vertex or per-pixel lighting hardware to generate a variety of useful effects. One

method is to modulate the output of standard Blinn/Phong lighting computations by the result of evaluating a light vector dependent parametric texture map. By doing this, Equation (1) becomes:

$$I = k_a I_a + k_d I_d(N \cdot L) + k_s I_s(N \cdot H)^n PTM(u, v, L_u, L_v) \quad (3)$$

5 where  $PTM$  is the parametric texture map function as in Equation (2) and  $L_u$  and  $L_v$  represent the projection of the light source vector  $L$  as above. In the trivial case where the parametric texture map varies spatially but is constant for all incident light directions, Equation (3) becomes functionally equivalent to specular maps or gloss maps which are well known in the art. The dependence on light direction that is not available with conventional specular maps  
10 could be used to reproduce effects such as shadowing where texels do not exhibit specular highlights due to shadowing by small scale surface variations.

Another modification is to utilize parametric texture maps to simulate off-specular and Fresnel effects. Many surfaces exhibit increased specularity for low angles of incidence due to decreases in the apparent roughness. This can be reproduced using the above  
15 technique by modulating the specular contribution with a parametric texture map whose magnitude increases as the incident angle approaches grazing. Incident illumination dependence can also be used to approximate Fresnel effects which are well known. Fresnel effects are important for many surfaces, such as metal and glass, where reflectance increases greatly for grazing incident illumination angles. The reflectance of glass and other dielectrics  
20 are low when the incident direction is nearly equivalent to the surface normal. Metals also exhibit Fresnel reflectance including wavelength dependence. As a result, the color of reflected light changes as the incident angle changes due to reflectance variations for different wavelengths. These dependencies can be approximated by the polynomial function stored at each texel. The combination of these illumination dependent effects can all be  
25 stored in a single parametric texture map. In addition, since texel polynomials in the

parametric texture map are independent, these properties can vary across the surface. As a result, a parametric texture map can represent different materials in a single texture map.

Further still, anisotropic materials, which are well known, can be modeled using parametric texture maps. An anisotropic material is one in which the reflectance function has rotational variation about the surface normal. Examples include fur and hair, brushed materials, and burnished metals. Conventional techniques to render anisotropic materials define normal planes for surface points aligned with the direction of anisotropy. The projection of the incident light direction into the normal plane is then used in the lighting calculations. According to the present technique, the function for each texel in the texture map is based on the magnitude of the incident light direction projected into the normal plane. The result of evaluating the parametric texture map is then modulated with calculated specular lighting so that specular highlights occur only in regions where the incident light direction aligns with the direction of anisotropy. Unlike some conventional methods, this technique allows anisotropic materials to be rendered under perspective views with local light sources.

In contrast to well known conventional results, a wide range of new effects can also be achieved. This is made possible by using the parametric texture map to represent any desired data since the projected vector can be assigned by the user directly if desired. Users can choose to leave one of the parameters constant for representing one-dimensional (1D) functions. For example, some time variant functions can be represented in this way such as a photographic sequence of a scene as time varies during the day. One limitation to this effect is that it has difficulty handling scenes that contain abrupt changes from frame to frame.

In addition, instead of using the two axes of standard 2D texture mapping to encode spatial color variations, the two texture coordinates of the texture map can be used as parameters for a bivariate function to represent the reflectance properties of the object. The

present technique has the additional advantage over conventional techniques that instead of storing simple color values in the texture map, the present technique can store a function of two variables. This function can be controlled by the light, eye point, half-angle, or other vectors as desired. Therefore, the present technique can be used to represent and render objects with full four dimensional reflectance functions instead of only two as with conventional techniques. Also, unlike conventional techniques, the present technique can reproduce effects of local light or viewpoints.

Further still, the present technique can be utilized in volume rendering. Instead of simply storing 2D parametric texture maps, volume rendering makes use of 3D parametric texture maps. This can be visualized as stacks of 2D texture maps. Similar to above, for volume rendering, function coefficients are stored for every texel. One can then render the volume using 3D texture mapping hardware by evaluating the parametric texture map at every texel to determine voxel, unit volume element, color values. Additionally, an extra function can be used to represent the transparency of each voxel. The parameters for the transparency function are specified by the user to allow voxels with the desired properties to be more or less opaque than other voxels.

One of the difficulties with using traditional bump maps to display surface detail is filtering them to match a particular sampling rate. The naïve approach of using MIP maps results in smoothing the surface and thereby removing the bumps themselves. This defeats the purpose of the bump map. Upon review, one will realize that the function that one would like to integrate over is the light radiated from the bumpy patch rather than the surface perturbations themselves. Since parametric texture maps are image-based representations, one can achieve the filtering effect by MIP mapping the coefficients A1-A6 themselves. Support for such a conclusion can be realized by considering the following equation where one is integrating over a patch  $\Omega$  with  $n$  samples:

$$\frac{1}{n} \sum_{i,j \in \Omega} L_{r,g,b} ((A1 - 6)(u_i, v_j)) = L_{r,g,b} \left( \frac{1}{n} \sum_{i,j \in \Omega} (A1 - 6)(u_i, v_j) \right). \quad (4)$$

Inspection reveals that the light dependence is linear in the coefficients. According to Equation (4), accumulating coefficients over a patch and then evaluating the sum is equivalent to accumulating the colors each texel contributes when independently evaluated.

- 5 Not only does this allow one to MIP map the coefficients directly, but anisotropic filtering techniques that yield improved image quality such as footprint assembly can also be supported. Therefore there are two options for interpolation with parametric texture maps.
- First, one can evaluate the polynomial at all sample points and then interpolate the calculated values. Second, one can interpolate the polynomial coefficients and subsequently evaluate
- 10 the interpolated polynomial. Whether one option is preferred over the other will depend on the particular implementation.

While the invention has been illustrated and described by means of specific embodiments, it is to be understood that numerous changes and modifications may be made therein without departing from the spirit and scope of the invention as defined in the

- 15 appended claims and equivalents thereof.

**CLAIMS**

What is claimed is:

1. A computer graphics system including a buffer, a graphics processor, a frame buffer, and a display, the system comprising:
  - a parametric texture map connected to the graphics processor, the parametric texture map holds a set of parameters for an equation that defines a surface structure in a manner in which the appearance of the surface structure varies in response to a user-defined vector.
2. The system as defined in claim 1, wherein the parametric texture map exhibits reflectance variations for different incident illumination angles thereby approximating off-specular and Fresnel effects.
3. The system as defined in claim 1, wherein the parametric texture map is based on the magnitude of the projection of the incident light direction in the normal plane with the result of the evaluation of the parametric texture map being then modulated with calculated specular lighting so that specular highlights occur only in regions where incident light directions align with a direction of anisotropy thereby approximating anisotropic effects.
4. The system as defined in claim 1, wherein at least one of the parameters of the parametric texture map is chosen to be constant for representing a one-dimensional function.
5. The system as defined in claim 1, wherein none of the parameters of the parametric texture map are chosen to encode spatial color variation.

6. The system as defined in claim 1, wherein the parametric texture map is  
2 multiplied with each channel in a complete RGB texture map.
  
7. The system as defined in claim 1, wherein the parameters are coefficients that  
2 are MIP mapped to filter the parametric texture map to match a specified sampling rate.
  
8. The system as defined in claim 1, wherein the effect of the parametric texture  
2 map is attenuated at the transition from front facing to back facing.
  
9. The system as defined in claim 1, wherein the equation that defines the surface  
2 structure is a polynomial equation.
  
10. The system as defined in claim 1, wherein the equation that defines the surface  
2 structure is a non-polynomial equation.
  
11. A method of rendering 3D objects using a parametric texture map in a  
2 computer graphics system, the method comprising the steps of:  
obtaining coefficients for an equation that defines a surface structure in a manner in  
4 which the appearance of the surface structure varies in response to a user-defined vector; and  
storing the coefficients in the parametric texture map.
  
12. The method as defined in claim 11, further comprising the step of adapting the  
2 coefficients to exhibit increasing reflectance variations for different incident illumination  
angles thereby approximating off-specular and Fresnel effects.

13. The method as defined in claim 11, further comprising the step of adapting the  
2 coefficients to be based on the magnitude of the projection of the incident light direction in  
the normal plane with the result of the evaluation of the parametric texture map being then  
4 modulated with calculated specular lighting so that specular highlights occur only in regions  
where incident light directions align with a direction of anisotropy thereby approximating  
6 anisotropic effects.

14. The method as defined in claim 11, further comprising the step of adapting the  
2 coefficients so that at least one parameter of the parametric texture map is chosen to be  
constant for representing a one-dimensional function.

15. The method as defined in claim 11, further comprising the step of adapting the  
2 coefficients so that none of the parameters of the parametric texture map are chosen to  
encode spatial color variation.

16. The method as defined in claim 11, further comprising the step of multiplying  
2 each channel with the parametric texture map in a complete RGB texture map.

17. The method as defined in claim 11, further comprising the step of MIP  
2 mapping the coefficients to filter the parametric texture map to match a specified sampling  
rate.

18. The method as defined in claim 11, further comprising the step of attenuating  
2 the effect of the parametric texture map at the transition from front facing to back facing.

19. A computer graphics system including a buffer, a graphics processor, a frame  
2 buffer, and a display, the system comprising:

a parametric texture map connected to the graphics processor, the parametric texture  
4 map holds a set of parameters for an equation that defines a volume in a manner in which the  
appearance of a voxel varies in response to a user-defined vector.

20. A method of rendering 3D objects using a parametric texture map in a  
2 computer graphics system, the method comprising the steps of:

obtaining coefficients for an equation that models a volume in a manner in which the  
4 appearance of a voxel varies in response to a user-defined vector; and  
storing the coefficients in the parametric texture map.

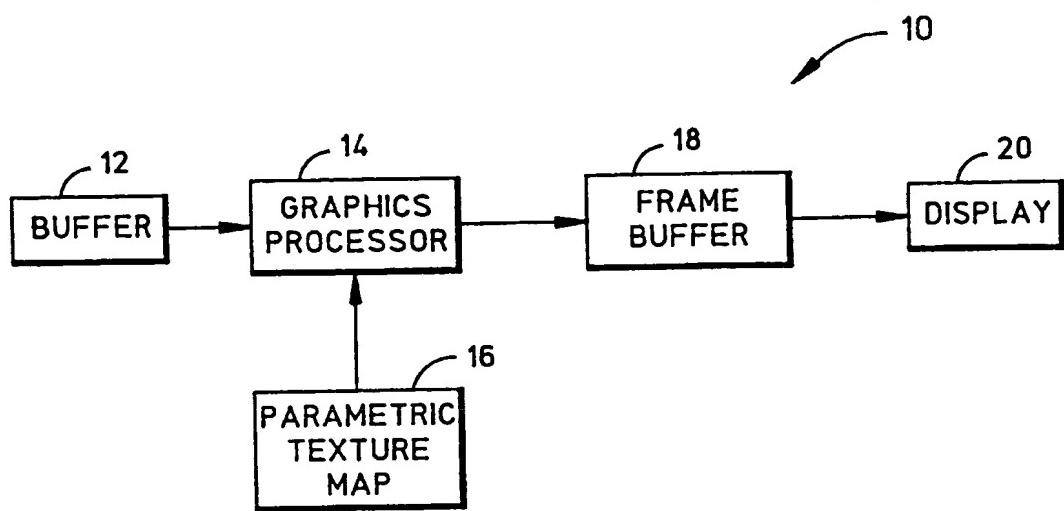


FIG. 1

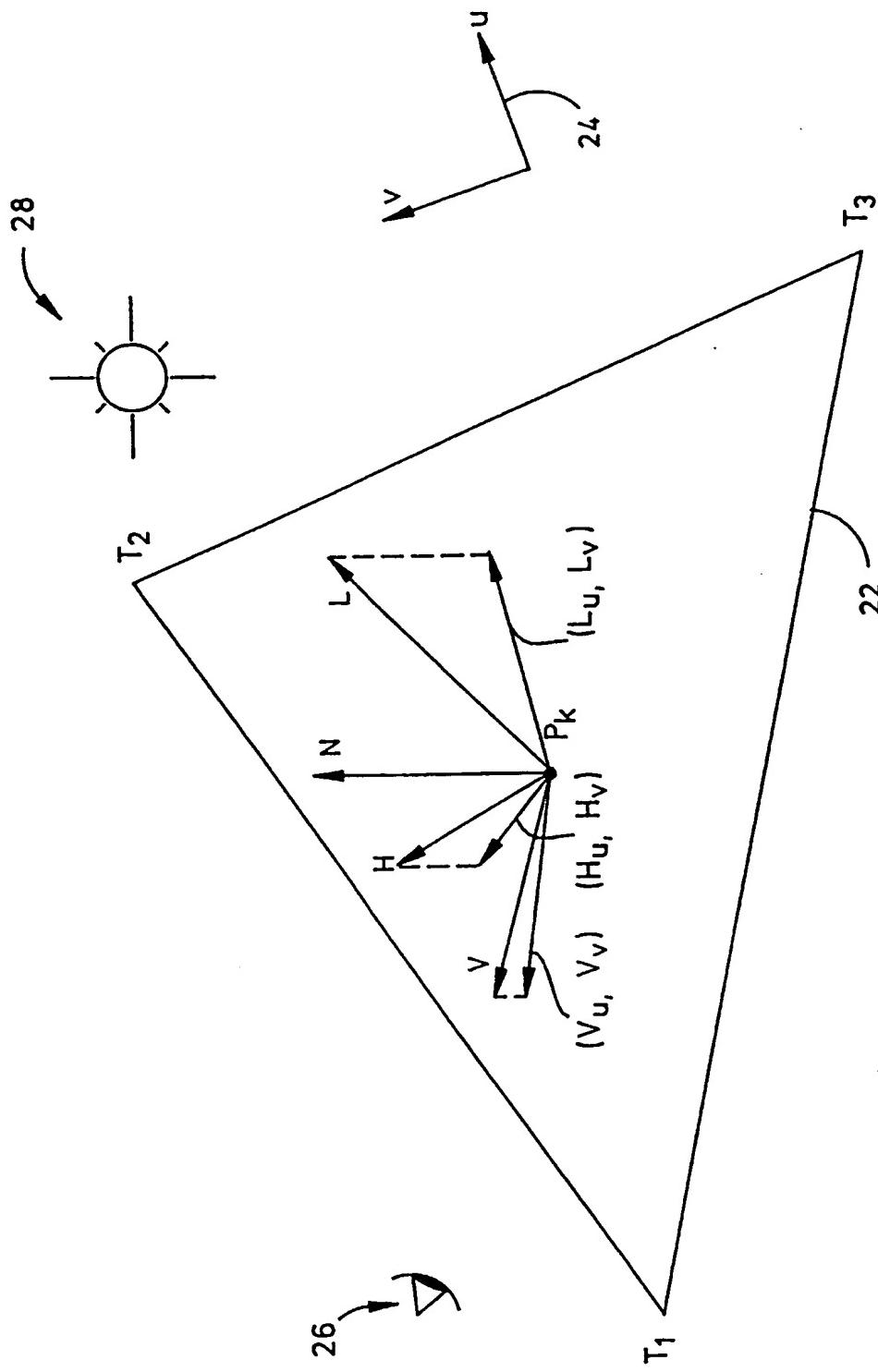


FIG. 2

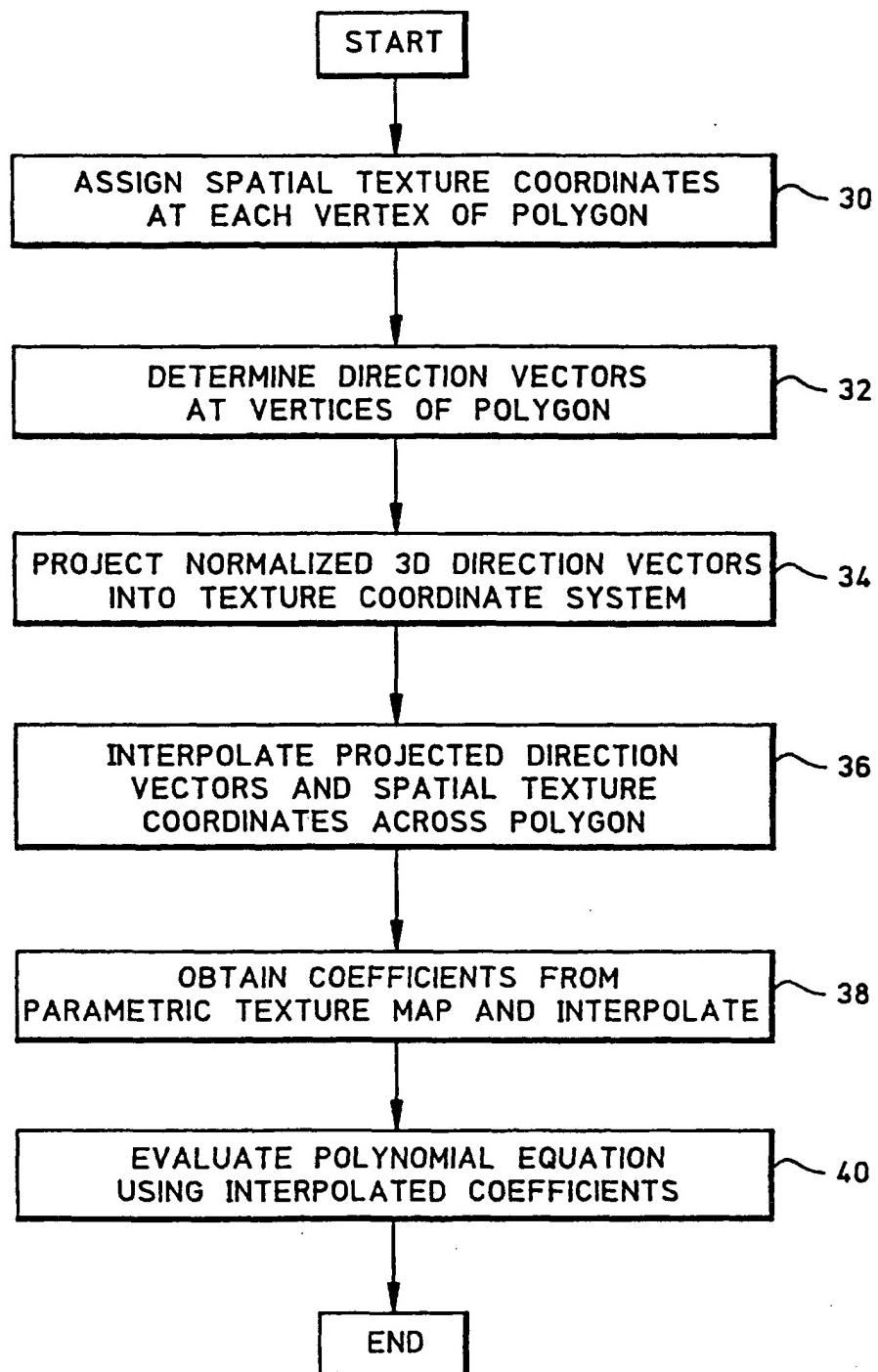


FIG. 3

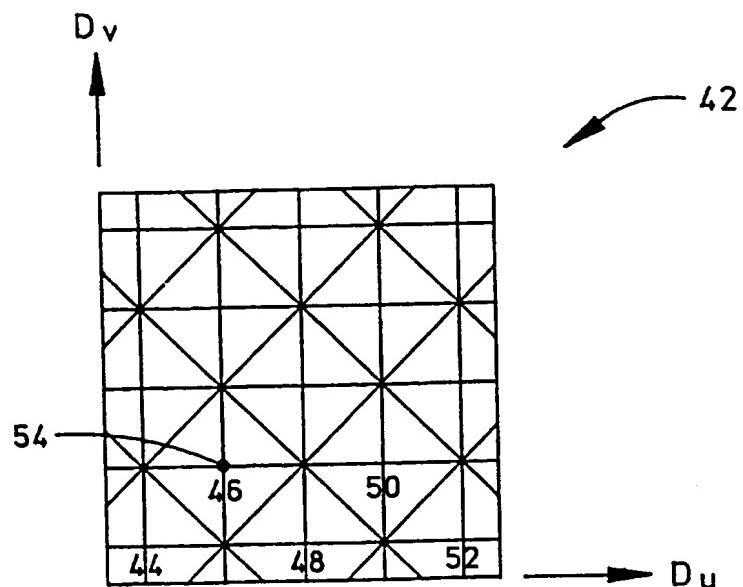


FIG. 4A

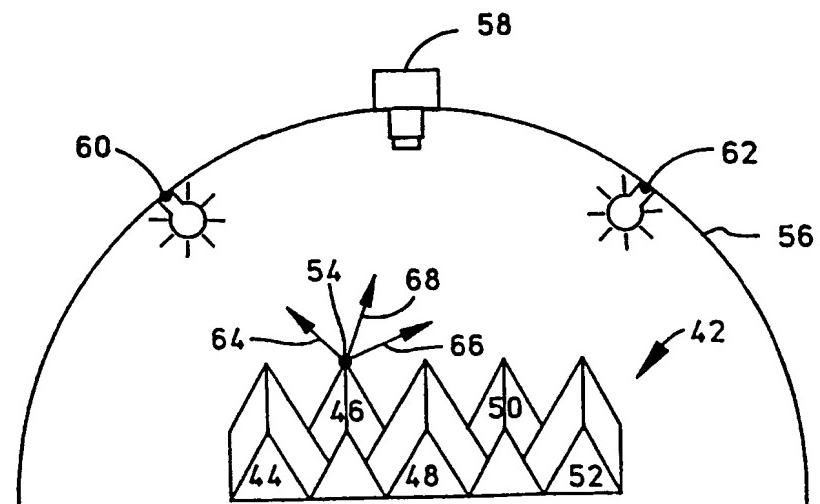


FIG. 4B

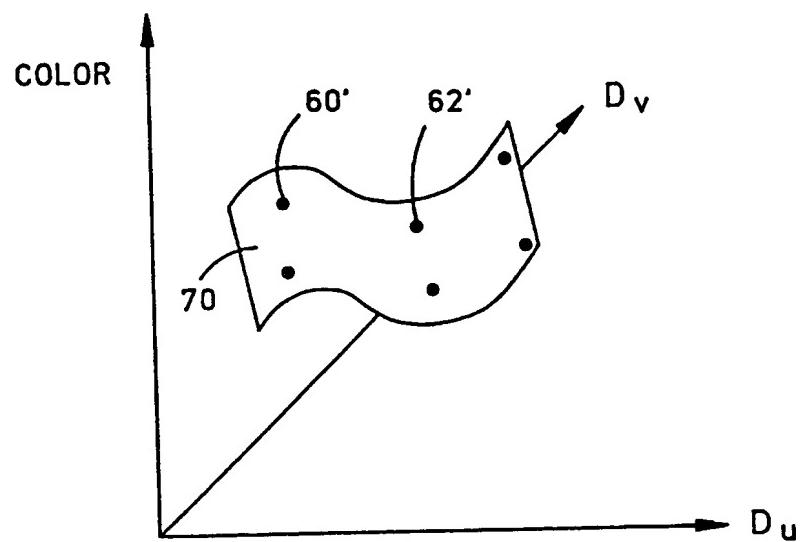


FIG. 4C